分布式检索题库

1611A班第01教学周期考题库

1、Solr 是什么？Solr它是一种开放源码的、基于 Lucene Java 的搜索服务器，易于加入到 Web 应用程序中。Solr 提供了层面搜索(就是统计)、命中醒目显示并且支持多种输出格式（包括XML/XSLT 和JSON等格式）。它易于安装和配置，而且附带了一个基于HTTP 的管理界面。

2、Solr 的表现优异的基本搜索功能？

也可以对它进行扩展从而满足企业的需要。Solr的特性包括：高级的全文搜索功能 专为高通量的网络流量进行的优化 基于开放接口（XML和HTTP）的标准 综合的HTML管理界面 可伸缩性－能够有效地复制到另外一个Solr搜索服务器 使用XML配置达到灵活性和适配性 可扩展的插件体系2. Lucene 是什么？Lucene是一个基于Java的全文信息检索工具包，它不是一个完整的搜索应用程序，而是为你的应用程序提供索引和搜索功能。Lucene 目前是 Apache Jakarta(雅加达) 家族中的一个开源项目。

/3、solr项目可以有多个核心(core，在solr自带的示例里有一个 collection1的core,放在 E:\dev\solr\solr-4.10.4\example\solr\collection1 )，可以在核心下面定义索引结构和存放索引

核心中必须要有两个配置文件？

   一个是 schema.xml,用来定义索引结构;

   一个是 solrconfig.xml，用来配置缓、扩展插件等

#### /4、Document

Document是Solr索引（动词，indexing）和搜索的最基本单元，它类似于关系数据库表中的一条记录，可以包含一个或多个字段（Field），每个字段包含一个name和文本值。字段在被索引的同时可以存储在索引中，搜索时就能返回该字段的值，通常文档都应该包含一个能唯一表示该文档的id字段。/5、Schema

Solr中的Schema类似于关系数据库中的表结构，它以schema.xml的文本形式存在在conf目录下，在添加文当到索引中时需要指定Schema，Schema文件主要包含三部分：****字段（Field）、字段类型（FieldType）、唯一键（uniqueKey）****

字段类型（FieldType）：用来定义添加到索引中的xml文件字段（Field）中的类型，如：int，String，date，

字段（Field）：添加到索引文件中时的字段名称

唯一键（uniqueKey）：uniqueKey是用来标识文档唯一性的一个字段（Feild），在更新和删除时用到

#### /6、Field

在Solr中，字段(Field)是构成Document的基本单元。对应于数据库表中的某一列。字段是包括了名称，类型以及对字段对应的值如何处理的一种元数据。/

7Solrconfig：

如果把Schema定义为Solr的Model的话，那么Solrconfig就是Solr的Configuration，它定义Solr如果处理索引、高亮、搜索等很多请求，同时还指定了缓存策略。

activeMQ应用场景：添加商品时需要将索引同步到数据库(这时候的manage-service充当消息的)当调用商品添加功能时,索引库并不会更新,此时如果对商品进行搜索,那么将无法获取到最新商品,所以需要当商品添加时,告诉索引库及时更新,此时activeMQ就充当了消息中间健的作用,当商品更新时,他就是消息生产者,通过activeMQ传达给消息消费者进行索引更新

solrcloud:solrcloud是solr提供的,基于solr和zookeeper的分布式搜索方案

应用场景 :1) 当需要大规模,容错,分布式索引和检索能力时使用solrcloud;

1. 当索引量很大,搜索请求并发很高时,同样需要使用solrcloud来满足这些需求;

solrcloud搭建: 1).复制4个单机版solr服务对应的tomcat,并分别改变端口(每一个上面都有三个端口要改);   2)复制4个solrhome,每个solr实例对应一个solrhome   3).修改solrhome下的solr,xml文件(改一下ip和端口)   4).设置tomcat启动参数,在每个tomcat目录下的 bin/catalina.sh添加一下内容:JAVA\_OPTS="-DzkHost=**192.168.242.138:2281,192.168.242.138:2282,192.168.242.138:2283**"  5).将solr文件中配置文件上传到zookeeper中,进行统一管理(使用/usr/local/solr-4.10.3/example/scripts/cloud-scripts下的zkcli.sh命令将/usr/local/solrcloud/solrhome8280/collection1/conf目录上传到zookeeper进行配置。):./zkcli.sh-zkhost192.168.242.139:2281,192.168.242.139:2282,192.168.242.139:2283-cmdupconfig -confdir /usr/local/solrcloud/solrhome8280/collection1/conf-confnamemyconf  6).使用zookeeper自带的 zkCli.sh命令连接zookeeper集群，查看上传的配置文件: **./zkCli.sh -server 192.168.242.138:2281   7)启动所有solr服务**

1. **创建collection:创建新的集群,名称为collection2,集群中有四个solr节点,将集群分为两片,每片两个副本: http://192.168.242.138:8280/solr/admin/collections?action=CREATE&name=collection2&numShards=2&replicationFactor=2      删除名称为collection1的集群:   <http://192.168.242.139:8280/solr/admin/collections?action=DELETE&name=collection1>**

9、安装前准备

solr依赖java 8 运行环境，所以我们先安装java。如果没有java环境无法启动solr服务，并且会看到如下提示：

1. [root@localhost solr-6.1.0]# ./bin/solr start -e cloud -noprompt /\*运行solr服务\*/
2. Java not found, or an error was encountered when running java.
3. A working Java 8 is required to run Solr!
4. Please install Java 8 or fix JAVA\_HOME before running this script.
5. Command that we tried: 'java -version'
6. [root@localhost ~]# java -version /\*检测是否已安装java\*/
7. [root@localhost ~]#  yum install -y java /\*安装java\*/

2、下载&安装&启动&停止

下载&安装：下载安装包，然后直接解压到指定目录即可。

1. [root@localhost ~]# wget http://ftp.meisei-u.ac.jp/mirror/apache/dist/lucene/solr/6.1.0/solr-6.1.0.tgz
2. [root@localhost ~]# tar zxvf solr-6.1.0.tgz -C /usr/local/
3. [root@localhost ~]# cd /usr/local/solr-6.1.0
4. [root@localhost solr-6.1.0]# ls
5. bin  CHANGES.txt  contrib  dist  docs  example  licenses  LICENSE.txt  LUCENE\_CHANGES.txt  NOTICE.txt  README.txt  server
6. [root@localhost solr-6.1.0]# ./bin/solr -help /\*查看solr命令帮助\*/
7. [root@localhost solr-6.1.0]# ./bin/solr status /\*查看solr服务启动状态\*/

**10、**启动solr服务

1. [root@localhost solr-6.1.0]# ./bin/solr start -e cloud -noprompt /\*启动solr服务\*/
2. Welcome to the SolrCloud example!
3. Starting up 2 Solr nodes for your example SolrCloud cluster.
4. /\*省略中间部分\*/
5. SolrCloud example running, please visit: http://localhost:8983/solr /\*浏览器打开这个url可以看到solr的运行情况\*/
6. [root@localhost solr-6.1.0]# ps aux | grep solr /\*可以看到solr服务进程\*/
7. [root@localhost solr-6.1.0]# ./bin/solr start -help /\*查看启动命令帮助\*/

**11.**停止solr服务

1. [root@localhost solr-6.1.0]# ./bin/solr stop -all
2. [root@localhost solr-6.1.0]# ./bin/solr start -help /\*查看停止命令帮助\*/

生成索引数据

solr使用./bin/post命令生成索引数据。

### 索引

Solr/Lucene采用的是一种反向索引，所谓****反向索引****：就是从关键字到文档的映射过程，保存这种映射这种信息的索引称为反向索引

* 左边保存的是字符串序列
* 右边是字符串的文档（Document）编号链表，称为倒排表（Posting List）

### 索引创建

假设有如下两个原始文档：  
文档一：Students should be allowed to go out with their friends, but not allowed to drink beer.  
文档二：My friend Jerry went to school to see his students but found them drunk which is not allowed.

****一：把原始文档交给分词组件(Tokenizer)****  
分词组件(Tokenizer)会做以下几件事情(这个过程称为：Tokenize)，处理得到的结果是词汇单元（Token）

1. 将文档分成一个一个单独的单词
2. 去除标点符号
3. 去除停词(stop word)
   * 所谓停词(Stop word)就是一种语言中没有具体含义，因而大多数情况下不会作为搜索的关键词，这样一来创建索引时能减少索引的大小。英语中停词(Stop word)如：”the”、”a”、”this”，中文有：”的，得”等。不同语种的分词组件(Tokenizer)，都有自己的停词(stop word)集合。经过分词(Tokenizer)后得到的结果称为词汇单元(Token)。上例子中，便得到以下****词汇单元(Token)****：

"Students"，"allowed"，"go"，"their"，"friends"，"allowed"，"drink"，"beer"，"My"，"friend"

****二：词汇单元(Token)传给语言处理组件(Linguistic Processor)****  
语言处理组件(linguistic processor)主要是对得到的词元(Token)做一些语言相关的处理。对于英语，语言处理组件(Linguistic Processor)一般做以下几点：

1. 变为小写(Lowercase)。
2. 将单词缩减为词根形式，如”cars”到”car”等。这种操作称为：stemming。
3. 将单词转变为词根形式，如”drove”到”drive”等。这种操作称为：lemmatization。

语言处理组件(linguistic processor)处理得到的结果称为****词(Term)****，例子中经过语言处理后得到的词(Term)如下：

"student"，"allow"，"go"，"their"，"friend"，"allow"，"drink"，"beer"，"my"，"friend"，"jerry"，"go"，"school"，"see"，"his"，"student"，"find"，"them"，"drink"，"allow"。

经过语言处理后，搜索drive时drove也能被搜索出来。****Stemming 和 lemmatization的异同：****

****一：对查询内容进行词法分析、语法分析、语言处理****

1. 词法分析：区分查询内容中单词和关键字，比如：english and janpan，”and”就是关键字，”english”和”janpan”是普通单词。
2. 根据查询语法的语法规则形成一棵树

****根据查询语句与文档的相关性，对结果进行排序****

我们把查询语句也看作是一个文档，对文档与文档之间的相关性（relevance）进行打分（scoring），分数高比较越相关，排名就越靠前。当然还可以人工影响打分，比如百度搜索，就不一定完全按照相关性来排名的。

如何评判文档之间的相关性？一个文档由多个（或者一个）词（Term）组成，比如：”solr”， “toturial”，不同的词可能重要性不一样，比如solr就比toturial重要，如果一个文档出现了10次toturial，但只出现了一次solr，而另一文档solr出现了4次，toturial出现一次，那么后者很有可能就是我们想要的搜的结果。这就引申出权重（Term weight）的概念。

****权重****表示该词在文档中的重要程度，越重要的词当然权重越高，因此在计算文档相关性时[影响力](http://www.amazon.cn/gp/product/B0044KME2E/ref=as_li_qf_sp_asin_il_tl?ie=UTF8&tag=importnew-23&linkCode=as2&camp=536&creative=3200&creativeASIN=B0044KME2E" \o "影响力" \t "http://www.importnew.com/_blank)就更大。通过词之间的权重得到文档相关性的过程叫做****空间向量模型算法(Vector Space Model)****